UNCLASSIFIED

AD NUMBER

AD814249

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 5 Aug 1963. Other requests shall be referred to Center for Naval Analysis, Operations Evaluation Group, Washington, DC.

AUTHORITY

CNA D/N ltr, 13 Sep 1971

THIS PAGE IS UNCLASSIFIED

AD814249

FILE COPY

OEG

This is an interim report of continuing research. It does not necessarily represent the opinion of OEG or the U. S. Navy. It may be modified or withdrawn at any time.

THIS DOCUMENT IS SUBJECT TO SPECIAL EXPORT CONTROLS AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR FOREIGN NATIONALS MAY BE MADE ONLY WITH PRIOR APPROVAL OF OCNO (Op03)

DDC
MAY 25 1967
C

Interim Research Memorandum

OPERATIONS EVALUATION GROUP

Center for Naval Analyses
WASHINGTON 25, D. C.

11 5 Aug[redacted] [redacted]63

12 35p.

9 INTERIM RESEARCH MEMO[redacted]

~~OPERATIONS EVALUATION~~ GROUP

6 AIR-TO-GROUND GUNNERY SIMULATION;
OEG COMPUTER PROGRAM 18-63P.

10 [redacted] P. E. DePoy [redacted] W. Sanjour

14 OEG-IRM-43

This is an interim report of continuing research. It does not necessarily represent the opinion of OEG or the U.S. Navy. It may be modified or withdrawn at any time.

(270 700)

INTERIM ~~RESEARCH~~ MEMORANDUM
IRM-43

ABSTRACT

~~This~~ The memorandum presents a usage manual for an IBM 7090 computer program which uses a Monte Carlo simulation to determine the probability of destroying a rectangular target with air-to-ground gunnery. The effect of correlation of successive aim points is considered. It is assumed that the aim point is distributed with a bivariate normal aiming error and that the individual rounds are distributed with an independent bivariate normal ballistic dispersion. A flow chart, a listing of the FORTRAN program and a sample problem are included.

I. INTRODUCTION

Hitherto, models for air-to-ground gunnery have been of two kinds. The first assumes that the aim points for successive rounds are uncorrelated and independent, i.e. that the correlation of successive aim points is zero. The second model is the salvo type which assumes that successive aim points are completely correlated. It is well known that both of these models are unsatisfactory descriptions of air-to-ground gunnery since the aim point does wander and the correlation is neither 0 nor 1, but some intermediate value. These two models have been used because the more general case defies simple mathematical analyses. However, with the introduction of high speed computers and Monte Carlo techniques, it is possible to simulate the more general case.

II. GENERAL DESCRIPTION

This program is a Monte Carlo simulation for use on the IBM 7090 computer. It determines the probability of destroying a rectangular target with air-to-ground gunnery on the basis of the following assumptions:

a. The aiming error and the ballistic dispersion are bivariate normal and are independent in the range coordinate (along the flight of path of the aircraft) and the deflection coordinate (normal to the flight path in the horizontal plane).

b. There is a conditional kill probability (probability of kill given a hit) associated with each round, and there is no cumulative damage effect.

c. The correlation coefficient between the n^{th} and m^{th} round is $a^{|n-m|}$ in the range coordinate and $b^{|n-m|}$ in the deflection coordinate where a and b represent the correlation constants (Note: when a and b are zero or one, the process degenerates into the independent or salvo cases, respectively).

d. The mean aim point is at the center of the target which is rectangular and is located with one axis parallel to the flight path.

In addition to the parameters specifying the target size, aiming and ballistic dispersion, correlation and conditional kill probability, the program inputs include the aircraft speed, the firing rate of the gun, the slant range from the target at the commencement of the firing run, the maximum number of rounds per pass, the number of Monte Carlo runs to be made and the increment in the number of rounds at which target kill probabilities should be determined.

III MATHEMATICAL DERIVATION

Let x_n and y_n be random variables denoting the position of the aim point of the n^{th} round in the plane normal to the line of sight between the aircraft and the target. Let α_n and β_n be independent Gaussian random variables with standard deviation 1 and mean 0. Let

$$x_1 = \sigma_x \alpha_1 \qquad (1a)$$

$$x_n = ax_{n-1} + \sigma_x (1-a^2)^{1/2} \alpha_n, \quad n \neq 1 \tag{1b}$$

$$y_1 = \sigma_y \beta_1 \tag{1c}$$

$$y_n = by_{n-1} + \sigma_y (1-b^2)^{1/2} \beta_n, \quad n \neq 1 \tag{1d}$$

where a, b, σ_x, and σ_y are constants.

Equation (1) may be rewritten so that

$$x_n = \sigma_x \left(a^{n-1} \alpha_1 + (1-a^2)^{1/2} \sum_{i=2}^{n} a^{n-i} \alpha_i\right) \tag{2a}$$

$$y_n = \sigma_y \left(b^{n-1} \beta_1 + (1-b^2)^{1/2} \sum_{i=2}^{n} b^{n-i} \beta_i\right) \tag{2b}$$

It can be seen from equation (2) that

$$E(x_n) = E(y_n) = 0$$

$$E(x_n^2) = \sigma_x^2$$

$$E(y_n^2) = \sigma_y^2$$

$$E(x_n x_m) = \sigma_x^2 a^{|m-n|}$$

$$E(y_n y_m) = \sigma_y^2 b^{|m-n|}$$

We therefore have a process in which;

- each aim point is normally distributed with mean zero and standard deviation σ_x, σ_y.
- the correlation coefficient between the n^{th} and m^{th} round is $a^{|n-m|}$ in the x coordinate and $b^{|n-m|}$ in the y coordinate.
- when a and b equal zero or one, the process degenerates into the independent or salvo cases, respectively.

IV. METHOD OF SOLUTION

The inputs for the program consist of the following parameters:

Address*	Symbol	Description
1	a	Correlation constant in the range coordinate $(0 \le a \le 1)$
2	b	Correlation constant in the deflection coordinate, $(0 \le b \le 1)$
3	σ_R	Standard deviation of aim error in the range coordinate (mils)
4	σ_D	Standard deviation of aim error in the deflection coordinate (mils)
5	β_R	Standard deviation of ballistic dispersion in the range coordinate (mils)
6	β_D	Standard deviation of ballistic dispersion in the deflection coordinate (mils)
7	S	Slant range at commencement of firing run (ft.)
8	R	Firing rate (rounds per minute)
9	c	Aircraft speed (knots)
10	N	Maximum number of rounds per pass
11	P	Conditional kill probability of a round
12	L	Target length (normal to flight path in range coordinate) (ft.)
13	W	Target width (normal to flight path in deflection coordinate) (ft.)
14	F	Number of Monte Carlo iterations
15	N_r	Number of dummy passes through random number generators
16	ΔN	Increment in the number of rounds at which target kill probabilities are determined.

*The parameter addresses are explained in appendix C, Data Subroutine.

Unless otherwise specified, the following parameters are assigned the designated values:

Address	Parameter	Value
1	a	0
2	b	0
9	c	0
11	P	1.0
14	F	1000
16	ΔN	1
15	N_r	0

The program flow chart and FORTRAN statements are shown in appendixes A and B. The subroutines are described in appendixes C, D and E. The solution is obtained in the following manner:

1. Before starting the first iteration for the first data set, n_r dummy passes are made through the random number generator.

2. For each Monte Carlo iteration:

   a. Given the standard deviations of aim error (σ_R, σ_D) and the aim point of the $(n-1)^{st}$ round, two Gaussian-distributed (mean = 0, standard deviation = 1) random numbers (α, γ) are selected, and and the aim point of the n^{th} round (R_n, D_n) is determined:

   $$R_n = aR_{n-1} + \sigma_R (1-a^2)^{1/2} \alpha \qquad (1)$$

   $$D_n = bD_{n-1} + \sigma_D (1-b^2)^{1/2} \gamma \qquad (2)$$

   For the first round ($n = 1$):

   $$R_1 = \sigma_R \alpha \qquad (3)$$

   $$D_1 = \sigma_D \gamma \qquad (4)$$

   b. Given the target dimensions (L, W), the firing rate (R), the aircraft speed (c) and the initial slant range (S), the half-target size (in mils) is determined for the time at which the n^{th} round is fired:

   $$1/2\, L_n = \frac{500L}{S - (n-1)c(\frac{60}{R})(1.688)} \qquad (5)$$

   $$1/2W_n = \frac{500W}{S - (n-1)c(\frac{60}{R})(1.688)} \qquad (6)$$

   c. The previously computed aim point is checked to determine if it is within three standard deviations of ballistic dispersion (β_R, β_D) from the target:

   $$|R_n| \le 1/2L_n + 3\beta_R \ ? \qquad (7)$$

   $$|D_n| \le 1/2W_n + 3\beta_D \ ? \qquad (8)$$

If the aim point is not within three standard deviations of ballistic dispersion from the target in either coordinate, it is assumed that the round misses the target and the aim point for the $(n + 1)^{st}$ round is computed (part a.).

d. If the aim point is within three standard deviations from the target in both coordinates, a Gaussian-distributed random number (δ) is selected, and the impact point of the round in the range coordinate is determined and checked to ascertain whether the round falls within the limits of the target in the range coordinates:

$$|R_n + \beta_R \delta| < 1/2\, L_n \ ? \tag{9}$$

If the round does not fall within the target limits, the aim point for the $(n + 1)^{st}$ round is computed (part a.).

e. If the nth round does fall within the target limits in the range coordinate, another Gaussian-distributed random number (ϵ) is selected and the impact point of the round in the deflection coordinate is determined. Another check is then made to determine if the round hits the target:

$$|D_n + \beta_D \epsilon| < 1/2 W_n \ ? \tag{10}$$

If the round does not hit the target, the aim point for the $(n + 1)^{st}$ round is determined (part a.).

f. If the n^{th} round does hit the target, a uniformly distributed (between 0 and 1) random number (PP) is selected and compared with the conditional kill probability to determine if the hit results in a target kill. if $P < PP$, the weapon does not destroy the target and the aim point of the $(n + 1)^{st}$ weapon is computed (part a.).

g. If $P \geq PP$, the n^{th} weapon does kill the target. The count (N_h) of the Monte Carlo iterations for which the target is destroyed is increased by one. In addition, a counter (JJ_i) for the next highest multiple (i) of the increment (ΔN) in the number of rounds for which the probability is to be determined is increased by one:

$$\begin{aligned} N_h + 1 &\rightarrow N_h \\ JJ_i + 1 &\rightarrow JJ_i \end{aligned} \tag{11}$$

When a round has resulted in a kill or when the maximum number of rounds (N) has been fired without killing the target, the entire process is repeated until F Monte Carlo iterations have been completed.

3. At the conclusion, the estimated probability of destroying the target with N rounds is determined:

$$P(N) = N_h/F \; . \tag{12}$$

The probability of destroying the target with j rounds is determined:

$$P(j) = \sum_{i=1}^{n} JJ_i/F \tag{13}$$

$$j = n \, \Delta N \qquad (n = 1, 2, \ldots N/\Delta N) \tag{14}$$

V. USER'S INSTRUCTIONS

Input flexibility has been attained by allowing the user to vary any or all of the parameters in a computer run. There is no programmed limit to the number of data sets which a user may submit in a run. The only restriction is that each data set must terminate with one blank card, and the last set in the run must terminate with two blank cards.

For each data set after the first, the user need submit only those parameter values in a set that are different from those in the previous set. This is accomplished by identifying each input parameter by its address.

VI SAMPLE PROBLEM

A target whose dimensions in the plane normal to the flight path are 10 feet x 10 feet is to be attacked by a 500 KTAS aircraft with a firing rate of 1000 rounds per minute. The aircraft commences firing at a slant range of 5000 feet and fires a single burst of 100 rounds at the target. The probability of kill given a hit is 0.2. The standard deviations of aiming error are 6 and 4 mils in range and deflection, respectively, and the standard deviations of ballistic dispersion are 2 and 1 mils in range and deflection. The correlation constants in both range and deflection are assumed to be 0.8. What is the probability of killing the target with the entire burst and with smaller bursts in increments of 10 rounds? A Monte Carlo simulation of 2000 iterations is selected. The input statements are as follows:

Address	Value
1	0.8
2	0.8
3	6
4	4
5	2
6	1
7	5000
8	1000

(Continued from preceding page.)

Address	Value
9	500
10	100
11	0.2
12	10
13	10
14	2000
15	0
16	10

The submittal form and output for this sample problem are shown in appendix F.

VII. KEYPUNCH INSTRUCTIONS

Two methods of punching the input parameters can be used. In the first method, one card is used for each parameter. The address of the parameter is punched first, is followed by a space and then the parameter value is punched. In the second method, the packed card method, several parameters and their addresses can be punched on a single card (as many as space permits). In each case, the address is followed by its parameter value which is in turn followed by the next address and value. Each address and parameter value must be separated by a space and must also be separated from the value preceding or the address following it.

VIII. OPERATOR'S INSTRUCTIONS

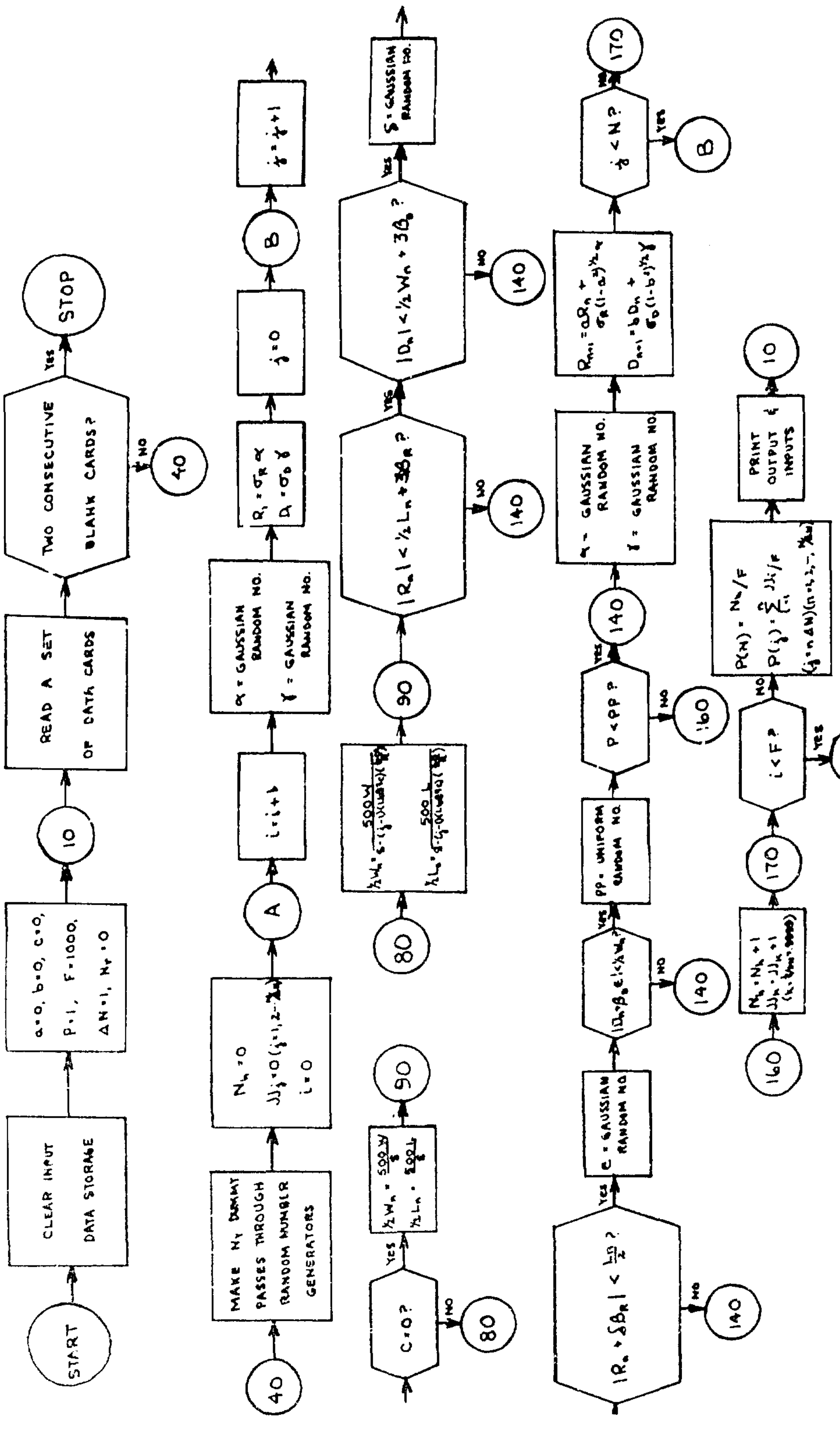
Run under control of the Bell System on the IBM 7090. No special instructions are required.

IX. TIMING

This program requires approximately 6 NF milliseconds running time per data set, where N is the number of rounds per pass and F is the number of Monte Carlo iterations.

APPENDIX A

FLOW CHART


START
CLEAR INPUT DATA STORAGE
10
READ A SET OF DATA CARDS
TWO CONSECUTIVE BLANK CARDS?
YES
STOP
NO
40
MAKE N_Y DUMMY PASSES THROUGH RANDOM NUMBER GENERATORS
A
B
GAUSSIAN RANDOM NO.
C=0?
80
90
140
160
170
UNIFORM RANDOM NO.
PRINT OUTPUT & INPUTS

APPENDIX B

FORTRAN STATEMENTS

```
      DIMENSION D(16), JJ(2000),FF(2000)
      EQUIVALENCE (D(1),A),(D(2),B),(D(3),SIGR),(D(4),SIGD),(D(5),BETAR
     1),(D(6),BETAD),(D(7),S),(D(8),R),(D(9),C),(D(10),FN),(D(11),P),
     2(D(12),FL), (D(13),W), (D(14),F), (D(16),DN)
      A=0.0
      B=0.0
      F=1000.0
      P=1.0
      DN=1.0
      C=0.0
   10 CALL DATA (D,IND)
      IF (IND) 40,40,20
   20 PRINT 30
   30 FORMAT (1H1)
      CALL ENDJOB
   40 PRINT 30
      II=D(15)
      DO 45 I=1,II
      CALL RANUMB (DUMMY)
   45 CALL GRNUMB (DUMMY)
      FLR2=500.0*FL/S
      WR2=500.0*W/S
      V=1.688*C
      VR=60.0*V/(S*R)
      N=FN/DN
      DO 50 I=1,N
   50 JJ(I)=0
      NH=0
      JJJ=FN
      II=F
      DO 170 I=1,II
      CALL GRNUMB(ALPHA)
      CALL GRNUMB (GAMMA)
      RC=SIGR*ALPHA
      DC=SIGD*GAMMA
      DO 150 J=1,JJJ
   60 IF (VR) 80,70,80
   70 WN2=WR2
      FLN2=FLR2
      GO TO 90
   80 WN2=WR2/(1.0-FLOATF(J-1)*VR)
      FLN2=FLR2/(1.0-FLOATF(J-1)*VR)
   90 IF(ABSF(RC)-FLN2-3.0*BETAR) 100,140,140
  100 IF (ABSF(DC)-WN2-3.0*BETAD) 110,140,140
  110 CALL GRNUMB (DELTA)
      IF (ABSF(RC+DELTA*BETAR)-FLN2) 120,140,140
  120 CALL GRNUMB (EPS)
      IF (ABSF(DC+EPS*BETAD)-WN2) 130,140,140
  130  CALL RANUMB(PP)
      JAB=J
      IF (PP-P) 160,140,140
```

```
140 CALL GRNUMB(ALPHA)
    CALL GRNUMB (GAMMA)
    RC=A*RC+SIGR*SQRTF(1.0-A**2)*ALPHA
150 DC=B*DC+SIGD*SQRTF(1.0-B**2)*GAMMA
    GO TO 170
160  NH=NH+1
    K=FLOATF(JAB)/DN+0.9999
    JJ(K)=JJ(K)+1
170 CONTINUE
    FH=FLOATF(NH)/F
    DO 180 I=2,N
    JJ(I)=JJ(I)+JJ(I-1)
180 FF(I)=FLOATF(JJ(I))/F
    FF(1)=FLOATF(JJ(1))/F
    PRINT 190, A,SIGR,BETAR,B,SIGD,BETAD
190 FORMAT (1H1, 49X, 11HCORRELATION,6X, 9HAIM ERROR, 4X, 9HBALLISTIC/
   133X, 10HCOORDINATE, 7X, 9HCONSTANTS, 8X, 6H(MILS), 4X, 17HDISPERSI
   2ON (MILS)//34X, 5HRANGE, 13X, F5.3, 10X, F5.1, 9X, F5.1/34X, 10HDE
   3FLECTION, 8X, F5.3, 10X, F5.1, 9X, F5.1)
    PRINT 200, C
200 FORMAT (/34X, 25H AIRCRAFT SPEED (KTAS)  =,F6.0)
    PRINT 205, R
205 FORMAT (34X, 25H FIRING RATE (RDS/MIN)  =,F6.0)
    PRINT 210, S
210 FORMAT (34X, 43H SLANT RANGE AT COMMENCEMENT OF FIRING RUN
   16H(FT) =,F7.0)
    KN=FN
    PRINT 215 , KN
215 FORMAT (34X, 25H NO. OF ROUNDS PER PASS =,I6)
    PRINT 220, P
220 FORMAT (34X, 31H CONDITIONAL KILL PROBABILITY =,F6.3)
    PRINT 225, FL
225 FORMAT (/34X, 23H TARGET LENGTH (FEET) =,F7.1)
    PRINT 230, W
230 FORMAT (34X, 23H TARGET WIDTH (FEET)  =,F7.1)
    KF=F
    PRINT 235, KF
235 FORMAT (/34X, 35H NUMBER OF MONTE CARLO ITERATIONS =,I6)
    NEMP=D(15)
    PRINT 240, NEMP
240 FORMAT (34X, 43H NUMBER OF EMPTY PASSES THROUGH RANDOM NO.
   112HGENERATORS =,I7)
    PRINT 245, FH
245 FORMAT (///34X, 29H PROBABILITY OF TARGET KILL =,F6.3)
    PRINT 250
250 FORMAT (///58X, 5H KILL/34X, 13HNO. OF ROUNDS,9X, 11HPROBABILITY)
    DO 260 I=1,N
    IDN=DN
    IR=I*IDN
260 PRINT 270, IR, FF(I)
270 FORMAT (39X, I5, 15X, F5.3)
    GO TO 10
    END
```

APPENDIX C

DATA SUBROUTINE

1. Introduction:

Many computer programs require the flexibility of varying any or all of the parameters in a computer run. Although FORTRAN is fairly flexible in its arithmetic and control statements, its input-output statements are quite rigid. In order to read cards for instance, considerable effort must be expended by the FORTRAN programmer in writing his input statements. This subroutine eliminates some of that tedium. The concept of a "data set" is used. A data set consists of a sequence of punched cards terminated by one blank card. A parameter deck for a computer run may consist of several data sets. Such a parameter deck is terminated by two blank cards.

2. Parameter Addresses:

The primary advantage of this subroutine over FORTRAN input statements results from the use of "parameter addresses." An address is a relative location in the computer memory. It is the subscript of an array or matirx. For example, in an array called X, the parameter value X_{53} would be located at address 53. By using the parameter addresses, a user of the program need submit only those parameter values in a data set that are different from those in the previous set.

Three types of addresses are permitted by this subroutine.

(1) A numeric address consisting of one to five characters, each of which is a digit 0 - 9. Such an address (n) refers to the n^{th} element in a specified array.

(2) An alpha address consisting of one to six characters, the first of which must be alphabetic (A-Z). The remaining may be alphabetic or numeric (A-Z or 0-9). Such an address refers to the n^{th} element in a specified array ($1 \leq n \leq 26$), where the first character of the address corresponds to n as the 26 letters of the alphabet correspond to the integers 1-26.

(3) A matrix address consisting of two or more numeric fields separated by commas. For example, the address 53, 47 refers to the element in the 53^{rd} row and the 47^{th} column of a two-dimensional matrix. There is no limit to the number of dimensions in a matrix address.

3. Input Card Format:

A standard submittal form (see attachment) has been designed for the analyst. This form provides for entering parameter values with their associated addresses. The user indicates blank cards to separate data sets. The keypunch operator has the option of punching one address and value per card, or, if the addresses are sequential, of punching one address and several values on a card.

Only columns 1-72 of a card are used. Each column must contain one of the following: a digit (0-9), a "+" or " - " sign or a dash, a letter (A-Z), a period, a comma, or a blank. Each punched card must contain one parameter address The address may start in column 1, or, if desired, may start in a later column, provided all columns before it are blank. The address is terminated by at least one blank column. Only one address is permitted on the card. Succeeding columns contain one or more parameter values, each separated by one or more blank columns. A value may be signed or unsigned. The length of the value field is variable. No blanks are permitted within a value field. A value may be punched with or without an exponent. An exponent is recognized by the presence of a plus or minus sign (or dash) between the fractional part and exponent part of the value. Decimal points (periods) may be punched in either the fractional or exponent parts of a value. If more than one value is punched on a card, those after the first will be entered at sequential addresses relative to the address of the first value.

4 Usage:

A data set is read by the use of the statement:

CALL DATA (X, I)

in a FORTRAN program for the IBM 7090. The argument X is the name of an array in the program. The argument I is an indicator set by the subroutine This indicator may be tested by the main program upon return from the subroutine. It will have a value of 0 or 1 or 2.

0: The subroutine has read a data set. The main program will normally proceed to operate on this data.

1: The subroutine has read the second blank card which terminates the parameter deck. The main program will normally terminate at this point.

2: The subroutine has read a "bad" data card. The main program may terminate the run, or ignore the card and return to the subroutine to read the rest of the data set.

If the cards to be read contain matrix addresses, additional arguments must be included in the FORTRAN calling statement:

CALL DATA (X, D_1, D_2, D_3, . . . , D_n, I)

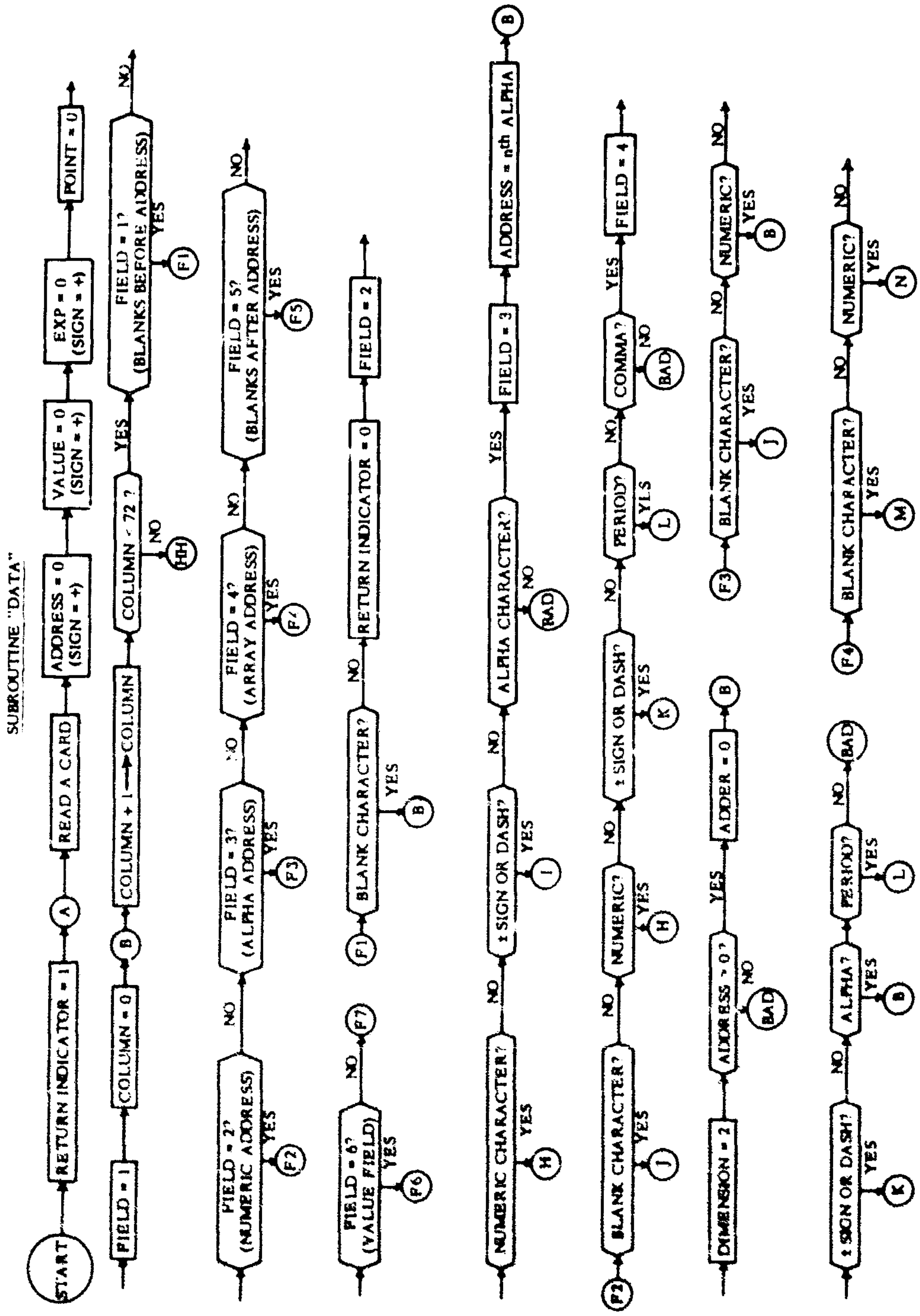
where D_i is the i^{th} dimension of the matrix X.

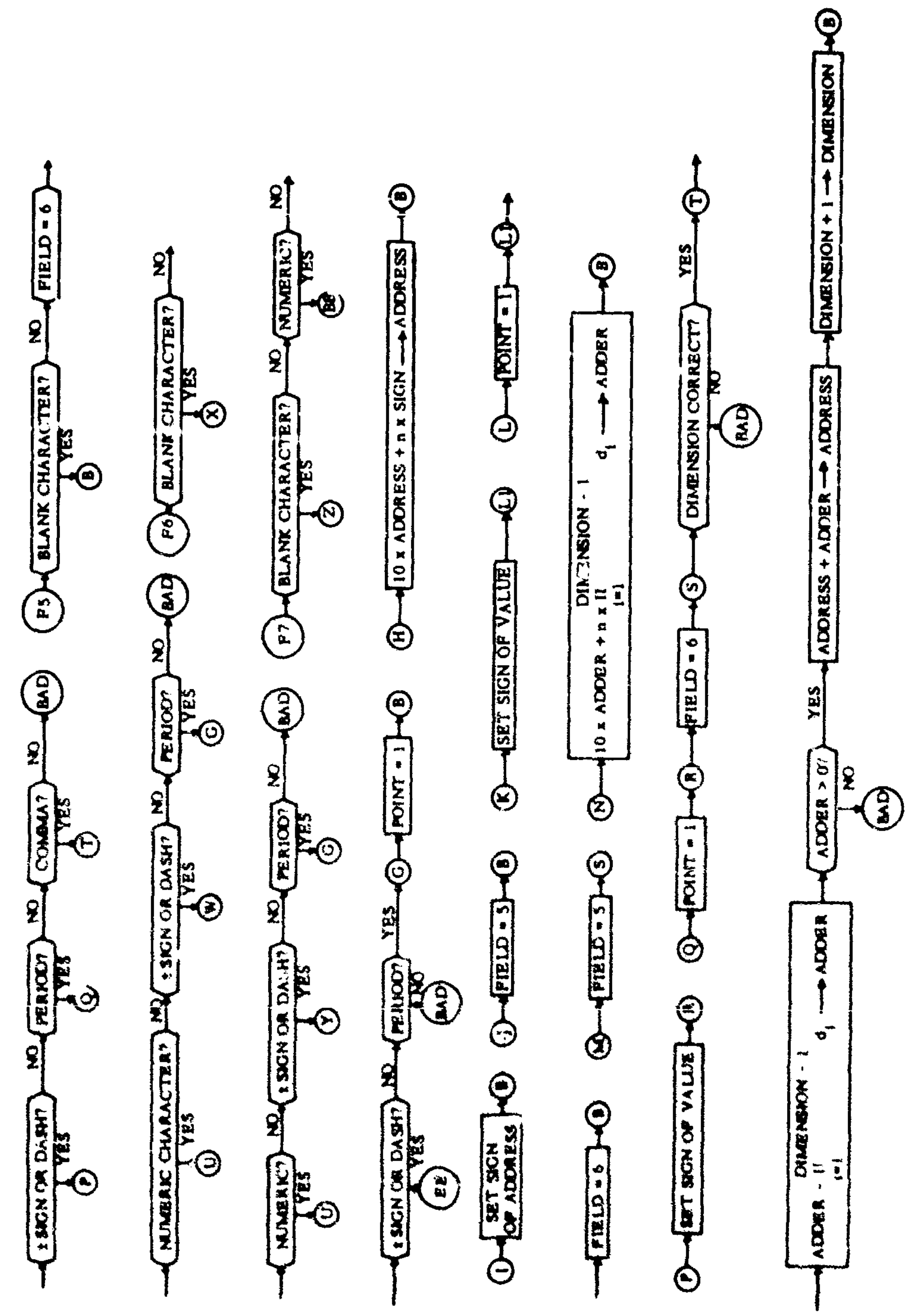
5 Method:

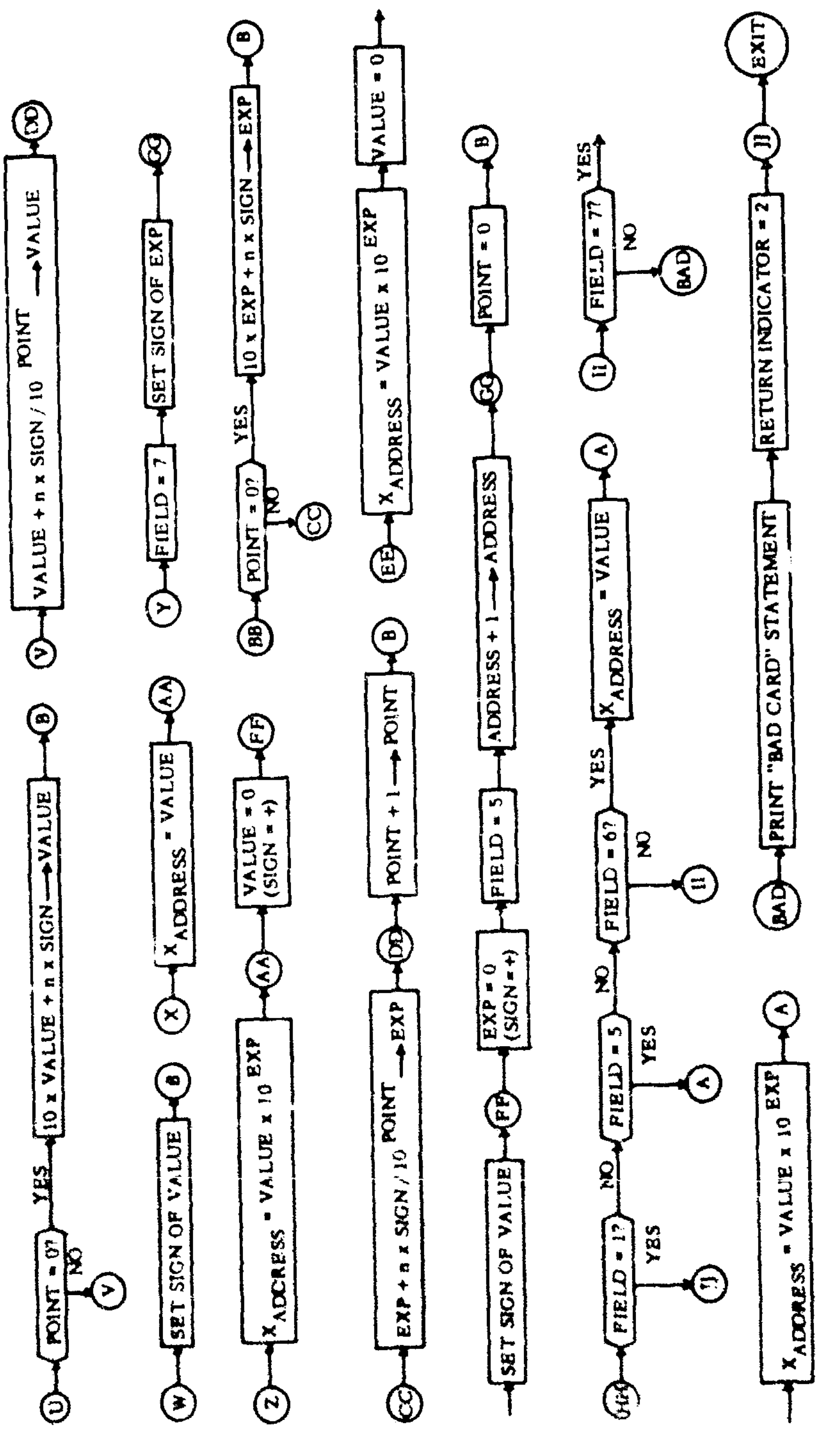
See the attached flow chart. DATA reads parameter values and loading addresses from cards. If sense switch 5 is up, it will read the values and addresses from tape (unit A2). It converts the values to floating point numbers, and stores them as elements of an array specified in the calling statement. The elements are specified by the addresses. If a card (or tape record) is read which contains non-permitted characters (see input card format above), DATA prints the statement "bad data card," followed by an image of the card itself.

6. Coding Information:

See the symbolic listing included in this appendix. DATA is written in the 7090 FAP language. It must be used in conjunction with the BELL system. It requires 401 words storage space.


SUBROUTINE "DATA"
START
RETURN INDICATOR = 1
A
READ A CARD
ADDRESS = 0 (SIGN = +)
VALUE = 0 (SIGN = +)
EXP = 0 (SIGN = +)
POINT = 0
FIELD = 1
COLUMN = 0
B
COLUMN + 1 → COLUMN
COLUMN > 72 ?
NO
HH
YES
FIELD = 1? (BLANKS BEFORE ADDRESS)
YES
F1
NO
FIELD = 2? (NUMERIC ADDRESS)
YES
F2
NO
FIELD = 3? (ALPHA ADDRESS)
YES
F3
NO
FIELD = 4? (ARRAY ADDRESS)
YES
F4
NO
FIELD = 5? (BLANKS AFTER ADDRESS)
YES
F5
NO
FIELD = 6? (VALUE FIELD)
YES
F6
NO
F7
F1
BLANK CHARACTER?
YES
B
NO
RETURN INDICATOR = 0
FIELD = 2
NUMERIC CHARACTER?
YES
H
NO
± SIGN OR DASH?
YES
I
NO
ALPHA CHARACTER?
NO
BAD
YES
FIELD = 3
ADDRESS = nth ALPHA
B
F2
BLANK CHARACTER?
YES
J
NO
NUMERIC?
YES
H
NO
± SIGN OR DASH?
YES
K
NO
PERIOD?
YES
L
NO
COMMA?
NO
BAD
YES
FIELD = 4
DIMENSION = 2
ADDRESS > 0?
NO
BAD
YES
ADDER = 0
B
F3
BLANK CHARACTER?
YES
J
NO
NUMERIC?
YES
B
NO
± SIGN OR DASH?
YES
K
NO
ALPHA?
YES
B
NO
PERIOD?
YES
L
NO
BAD
F4
BLANK CHARACTER?
YES
M
NO
NUMERIC?
YES
N
NO




POINT = 0?
YES
NO
10 x VALUE + n x SIGN → VALUE
VALUE + n x SIGN / 10 POINT → VALUE
SET SIGN OF VALUE
X ADDRESS = VALUE
FIELD = 7
SET SIGN OF EXP
X ADDRESS = VALUE x 10 EXP
VALUE = 0 (SIGN = +)
10 x EXP + n x SIGN → EXP
EXP + n x SIGN / 10 POINT → EXP
POINT + 1 → POINT
VALUE = 0
EXP = 0 (SIGN = +)
FIELD = 5
ADDRESS + 1 → ADDRESS
POINT = 0
FIELD = 1?
FIELD = 6?
FIELD = 7?
PRINT "BAD CARD" STATEMENT
RETURN INDICATOR = 2
EXIT
BAD

SYMBOLIC LISTING

```
        FAP
        ENTRY DATA
DATA    SXA X1,1
        SXA X2,2
        SXA X4,4
        CAL 1,4
        ADD CORE
        STO XLOC
        AXT 1,1
        SXA *+1,1
        CAL **,4
        ANA MASK
        TNZ *+2
        TXI *-4,1,1
        SXA EXIT,1
        TXI *+1,1,-1
        SXA *+1,1
        CLA **,4
        STA A1
        STA F1A
        STA II2
        AXT 1,1          RETURN INDICATOR = 1
A1      SXD **,1
A       TSX HHREAD,4     READ A CARD
        PZE CARD
        TRA EXIT
        TRA BAD
        STZ ADDRES       ADDRESS = 0
        STZ VALUE        VALUE = 0
        STZ EXP          EXP = 0
        STZ POINT        POINT = 0
        AXT 1,1          FIELD = 1
        SXA FIELD,1
        AXT 13,1
A2      TNX HH,1,1       COLUMN GT 72
        AXT 42,2
        SXA COLUMN,2
B       LXA COLUMN,2     COLUMN = COLUMN+1
        TNX A2,2,6
        SXA COLUMN,2
        LDQ CARD+12,1
        RQL 36,2
        PXD 0,0
        LGL 6
        STO CHARAC
        ORA FLOAT
        FAD FLOAT
        STO NUMB
        AXT 42,4
        CLA CHARAC
        CAS TABLE+42,4
        TRA *+2
        TRA *+3
        TIX *-3,4,1
```

```
        TRA BAD
        LXA FIELD,2
        TRA F1+1,2
        TRA F7          FIELD=7 (EXPONENT FIELD)
        TRA F6          FIELD=6 (VALUE FIELD)
        TRA F5          FIELD=5 (BLANKS AFTER ADDRESS)
        TRA F4          FIELD=4 (ARRAY ADDRESS)
        TRA F3          FIELD=3 (ALPHA ADDRESS)
        TRA F2          FIELD=2 (NUMERIC ADDRESS)
F1      TXH B,4,41      FIELD=1 (BLANKS BEFORE ADDRESS)
F1A     STZ **          RETURN INDICATOR = 0
        AXT 2,2         FIELD = 2
        SXA FIELD,2
        TXH H,4,31      NUMERIC CHARACTER
        TXH I,4,28      SIGN OR DASH
        TXL BAD,4,2
        AXT 3,2         ALPHA CHARACTER, FIELD = 3
        SXA FIELD,2
        TXI *+1,4,-2    ADDRESS = NTH ALPHA
        SXA ADDRES,4
        TRA B
F2      TXH J,4,41      BLANK CHARACTER
        TXH H,4,31      NUMERIC CHARACTER
        TXH K,4,28      SIGN OR DASH
        TXH BAD,4,2
        TXH L,4,1       PERIOD
        AXT 4,2         COMMA,FIELD = 4
        SXA FIELD,2
        AXT 2,2
        SXA DIMENS,2    DIMENSION = 2
        CLA ADDRES      TEST ADDRESS
        TZE BAD
        TMI BAD
F2A     STZ ADDER       ADDER=0
        TRA B
F3      TXH J,4,41      BLANK CHARACTER
        TXH B,4,31      NUMERIC CHARACTER
        TXH K,4,28      SIGN OR DASH
        TXH B,4,2       ALPHA CHARACTER
        TXH L,4,1       PERIOD
        TRA BAD
F4      TXH M,4,41      BLANK CHARACTER
        TXH N,4,31      NUMERIC CHARACTER
        TXH P,4,28      SIGN OR DASH
        TXH BAD,4,2
        TXH Q,4,1       PERIOD
        TRA T           COMMA
F5      TXH B,4,41      BLANK CHARACTER
        AXT 6,2         FIELD = 6
        SXA FIELD,2
        TXH U,4,31      NUMERIC CHARACTER
        TXH W,4,28      SIGN OR DASH
        TXH BAD,4,2
        TXH G,4,1       PERIOD
```

```
         TRA BAD
F6       TXH X,4,41      BLANK CHARACTER
         TXH U,4,31      NUMERIC CHARACTER
         TXH Y,4,28      SIGN OR DASH
         TXH BAD,4,2
         TXH G,4,1       PERIOD
         TRA BAD
F7       TXH Z,4,41      BLANK CHARACTER
         TXH BB,4,31     NUMERIC CHARACTER
         TXH EE,4,28     SIGN OR DASH
         TXH BAD,4,2
         TXL BAD,4,1
G        AXT 1,2         PERIOD, POINT = 1
         SXA POINT,2
         TRA B
H        LDQ ADDRES      ADDRESS = 10 X ADDRESS + N
         MPY H10
         XCA
         ACL CHARAC
         STO ADDRES
         TRA B
I        TXH B,4,30      + SIGN
         CLA ADDRES      SET SIGN OF ADDRESS
         SSM
         STO ADDRES
         TRA B
J        AXT 5,2         FIELD = 5
         SXA FIELD,2
         TRA B
K        TXH L1,4,30     + SIGN
         CLA VALUE       SET SIGN OF VALUE
         SSM
         STO VALUE
         TRA L1
L        AXT 1,2         POINT = 1
         SXA POINT,2
L1       AXT 6,2         FIELD = 6
         SXA FIELD,2
         TRA B
M        AXT 5,2         FIELD = 5
         SXA FIELD,2
         TRA S
N        LDQ ADDER       ADDER = 10 X ADDER + N X PROD
         MPY H10
         STQ ADDER
         TSX T1,4
         MPY CHARAC
         XCA
         ADD ADDER
         STO ADDER
         TRA B
P        TXH R,4,30      + SIGN
         CLA VALUE       SET SIGN OF VALUE
         SSM
```

```
        STO VALUE
        TRA R
Q       AXT 1,2           POINT = 1
        SXA POINT,2
R       AXT 6,2           FIELD = 6
        SXA FIELD,2
S       LXA EXIT,2        CHECK DIMENSION
        TXI *+1,2,-3
        PXA 0,2
        SUB DIMENS
        TNZ BAD
T       TSX T1,4          ADDER=ADDER-PROD
        CLA ADDER
        SUB PROD
        STO ADDER
        TZE BAD           CHECK ADDER
        TMI BAD
        ADD ADDRES
        STO ADDRES
        CLA DIMENS
        ADD H1
        STO DIMENS
        TRA F2A
T1      SXA T4,4          PROD = PRODUCT OF DIMENSIONS
        CLA H1
        STO PROD
        STA T3
        LXA DIMENS,2
        TXI *+1,2,-1
        LXA X4,4
T2      CAL T3
        ADD H1
        STA T3
T3      CLA **,4
        STA *+1
        LDQ **
        RQL 18
        MPY PROD
        STQ PROD
        TIX T2,2,1
T4      AXT **,4
        TRA 1,4
U       CLA POINT         TEST POINT
        TNZ V
        LDQ VALUE         VALUE = 10 X VALUE + N
        FMP DEC10
        SSP
        FAD NUMB
        LDQ VALUE
        LLS 0
        STO VALUE
        TRA B
V       LXA POINT,4       VALUE = VALUE + N/(10**POINT)
        CLA NUMB
```

```
        FDP DEC10
        XCA
        TIX *-2,4,1
        LDQ VALUE
        LLS 0
        FAD VALUE
        STO VALUE
        TRA DD
W       TXH B,4,30        + SIGN
        CLA VALUE         SET SIGN OF VALUE
        SSM
        STO VALUE
        TRA B
X       CLA XLOC          X(ADDRESS) = VALUE
        SUB ADDRES
        STA *+2
        CLA VALUE
        STO **
        TRA AA
Y       AXT 7,2
        SXA FIELD,2       FIELD = 2
        TXH GG,4,30       + SIGN
        CLA EXP           SET SIGN OF EXP
        SSM
        STO EXP
        TRA GG
Z       CLA XLOC          X(ADDRESS) = VALUE X 10**EXP
        SUB ADDRES
        STA Z1
        CLA DEC10
        LDQ EXP
        CALL EXP(3
        XCA
        FMP VALUE
Z1      STO **
AA      STZ VALUE         VALUE = 0
        TRA FF
BB      CLA POINT         TEST POINT
        TNZ CC
        LDQ EXP           EXP = 10 X EXP + N
        FMP DEC10
        SSP
        FAD NUMB
        LDQ EXP
        LLS 0
        STO EXP
        TRA B
CC      LXA POINT,4       EXP = EXP + N/(10**POINT)
        CLA NUMB
        FDP DEC10
        XCA
        TIX *-2,4,1
        LDQ EXP
        LLS 0
```

```
        FAD EXP
        STO EXP
DD      CLA POINT          POINT = POINT + 1
        ADD H1
        STO POINT
        TRA B
EE      CLA XLOC           X(ADDRESS) = VALUE X 10**EXP
        SUB ADDRES
        STA EE1
        CLA DEC10
        LDQ EXP
        CALL EXP(3
        XCA
        FMP VALUE
EE1     STO **
        PXD 0,0            VALUE = 0
        TXH *+2,4,30       + SIGN
        SSM                SET SIGN OF VALUE
        STO VALUE
FF      STZ EXP            EXP = 0
        AXT 5,2            FIELD = 5
        SXA FIELD,2
        CAL ADDRES         ADDRESS = ADDRESS + 1
        ADD H1
        SLW ADDRES
GG      STZ POINT          POINT = 0
        TRA B
HH      LXA FIELD,1
        TXL JJ,1,1         FIELD=1, EXIT
        TXL BAD,1,4
        TXL A,1,5          FIELD=5, READ ANOTHER CARD
        TXH II,1,6
        CLA XLOC           FIELD=6, X(ADDRESS) = VALUE
        SUB ADDRES
        STA *+2
        CLA VALUE
        STO **
        TRA A
II      TXH BAD,1,7
        CLA XLOC           FIELD=7,
        SUB ADDRES         X(ADDRESS) = VALUE X 10**EXP
        STA II1
        CLA DEC10
        LDQ EXP
        CALL EXP(3
        XCA
        FMP VALUE
II1     STO **
        TRA A
BAD     TSX HPRINT,4
        PZE PRINT,0,15
        AXT 2,1
II2     SXD **,1
X1      AXT **,1
```

```
X2     AXT **,2
X4     AXT **,4
EXIT   TRA **,4
MASK   OCT 777777700000
PRINT  BCD 3 BAD DATA CARD...
CARD   BSS 12
ADDRES HTR **
VALUE  HTR **
EXP    HTR **
POINT  HTR **
FIELD  HTR **
COLUMN HTR **
TABLE  OCT 60              BLANK
       OCT 0               0
       OCT 1               1
       OCT 2               2
       OCT 3               3
       OCT 4               4
       OCT 5               5
       OCT 6               6
       OCT 7               7
       OCT 10              8
       OCT 11              9
       OCT 20              + SIGN
       OCT 40              - SIGN
       OCT 14              DASH
       OCT 71              Z
       OCT 70              Y
       OCT 67              X
       OCT 66              W
       OCT 65              V
       OCT 64              U
       OCT 63              T
       OCT 62              S
       OCT 51              R
       OCT 50              Q
       OCT 47              P
       OCT 46              O
       OCT 45              N
       OCT 44              M
       OCT 43              L
       OCT 42              K
       OCT 41              J
       OCT 31              I
       OCT 30              H
       OCT 27              G
       OCT 26              F
       OCT 25              E
       OCT 24              D
       OCT 23              C
       OCT 22              B
       OCT 21              A
       OCT 33              PERIOD
       OCT 73              COMMA
```

```
CHARAC HTR **
DIMENS HTR **
 ADDER HTR **
 H10   HTR 10
 DEC10 DEC 10.0
 H1    HTR 1
 PROD  HTR **
 AMASK OCT 77777
 FLOAT OCT 233000000000
 NUMB  HTR **
 XLOC  HTR **
 CORE  OCT 100001
 JJ    SYN X1
       END
```

OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: ______________________________ **Date:** ______________

Program No. ____________ **Est. Time** ____________ **Classification** ____________

Special Instructions: ____

Address	Value	Address	Value	Address	Value	Address	Value

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ————b———— .

APPENDIX D

GRNUMB SUBROUTINE

1. Purpose:

GRNUMB provides a floating point pseudo-random number X. The distribution of successive values of X are Gaussian with a mean of zero and a standard deviation of one.

2. Method:

Consider the set of uniformly distributed pseudo-random numbers Y_i. GRNUMB generates a sequence of Y_i by the method of congruences:

$$Y_i = 2^{-35}(5^{15}2^{35}Y_{i-1}, \bmod 2^{35})$$

over the range $0 \leq Y_i < 1$. The variance of this uniform set is

$$\sigma_Y{}^2 = \int_0^1 (Y-1/2)^2 \, dY = 1/12 .$$

If X is the mean of any selection of m of the uniform numbers Y, the Central Limit Theorem states that the variable X approaches a normal distribution where m is sufficiently large. A satisfactory value for m is 30. Values of X are generated as a sequence of X_n, where n denotes the n^{th} entry to GRNUMB.

$$X_n = \sqrt{1/m\sigma_Y{}^2} \sum_{i=1}^{m} (Y_i{}^{-1/2}) = \sqrt{.4} \sum_{i=1}^{30} (Y_i{}^{-1/2})$$

where $Y_o = X_{n-1}$, and $X_o = 2^{-35}$. The variance of this normal set is 1.

3. Usage:

X is obtained by use of the statement:

CALL GRNUMB (X)

in a FORTRAN program for the IBM 7090.

4. Coding Information:

See the symbolic listing on the following page. GRNUMB is written in the 7090 FAP language. It requires 40 words storage space and 900 microseconds operating time.

SYMBOLIC LISTING

```
              FAP
              REM GRNUMB G. WESTLUND     18 JUNE 1962   (7Ø9Ø)
              REM GAUSSIAN DISTRIBUTED RANDOM NUMBER GENERATOR.
              REM ENTER VIA FORTRAN STATEMENT        CALL GRNUMB (X)
              REM SEQUENCE STARTS AT DEC 1, YIELDS X WITH STAND. DEV. =1.
              ENTRY GRNUMB
GRNUMB        SXA XX1, 1
              CLA 1, 4
              STA F
              AXT 3Ø, 1
              STZ NUM
C             LDQ NUMB
              MPY MULT
              STQ NUMB
              CLA NUMB
              SUB CHAR
              ARS 4
              ADD NUM
              STO NUM
              TIX C, 1, 1
              LDQ NUM
              MPY MAGIC
              LRS 27
              TZE D
              LRS 8
              CLA H125
              ADD H8
              LLS 8
              ALS 19
              TRA E
D             CLA H125
              ALS 27
E             STO NUM
              CLA H125
              LLS 27
              FAD NUM
F             STO **
XX1           AXT **, 1
              TRA 2, 4
NUM           HTR **
NUMB          DEC 1
MULT          DEC 3Ø517578125
CHAR          TIX Ø, Ø, Ø
MAGIC         DEC 0.3162278ØBØ
H8            DEC 8
H125          DEC 125
              END
```

APPENDIX E

RANUMB SUBROUTINE

1. Purpose:

RANUMB provides a floating point pseudo-random number X. Successive values of X are uniformly distributed over the range $0 \le X < 1$.

2. Method:

The generated value of X is a member of a set of pseudo-random numbers. This set is a sequence of X_n, where n denotes the n^{th} entry to RANUMB. The set is generated by the method of congruences:

$$N_n = (5^{15} N_{n-1}), \bmod 2^{35}$$

$$X_n = N_n / 2^{35}$$

where $N_o = 1$.

3. Usage:

X is obtained by use of the statement:

CALL RANUMB (X)

in a FORTRAN program for the IBM 7090.

4. Coding Information:

See the symbolic listing on the following page. RANUMB is written in the 7090 FAP language. It requires 15 words storage space and 41 microseconds operating time.

SYMBOLIC LISTING

```
        FAP
        REM RANUMB G. WESTLUND   18 JUNE 1962 (7Ø9Ø)
        REM UNIFORMLY DISTRIBUTED RANDOM NUMBER GENERATOR.
        REM ENTER VIA FORTRAN STATEMENT       CALL RANUMB (X)
        REM SEQUENCE STARTS AT DEC 1, YIELDS (Ø LE X LT 1).
        ENTRY RANUMB
RANUMB  CLA 1, 4
        STA C
        LDQ NUMB
        MPY MULT
        STQ NUMB
        PXD Ø, Ø
        RQL 9
        LGL 27
        ACL CHAR
        FAD CHAR
C       STO **
        TRA 2, 4
NUMB    DEC 1
MULT    DEC 3Ø517578125
CHAR    TIX Ø, Ø, Ø
        END
```

APPENDIX F

SAMPLE PROBLEM SUBMITTAL FORM

OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: John Doe Date: 26 July 1963

Program No. ______ Est. Time 4 min Classification Unclassified

Special Instructions: ______

Address	Value	Address	Value	Address	Value	Address	Value
1	0.8						
2	0.8						
3	6						
4	4						
5	2						
6	1						
7	5000						
8	1000						
9	500						
10	100						
11	0.2						
12	10						
13	10						
14	2000						
15	0						
16	10						
———	b———						
———	b———						

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ————b———— .

APPENDIX G

SAMPLE PROBLEM OUTPUT

COORDINATE	CORRELATION CONSTANTS	AIM ERROR (MILS)	BALLISTIC DISPERSION (MILS)
RANGE	0.800	6.0	2.0
DEFLECTION	0.800	4.0	1.0

AIRCRAFT SPEED (KTAS) = 500.
FIRING RATE (RDS/MIN) = 1000.
SLANT RANGE AT COMMENCEMENT OF FIRING RUN (FT) = 5000.
NO. OF ROUNDS PER PASS = 100
CONDITIONAL KILL PROBABILITY = 0.200

TARGET LENGTH (FEET) = 10.0
TARGET WIDTH (FEET) = 10.0

NUMBER OF MONTE CARLO ITERATIONS = 2000
NUMBER OF EMPTY PASSES THROUGH RANDOM NO. GENERATORS = 0

PROBABILITY OF TARGET KILL = 0.993

NO. OF ROUNDS	KILL PROBABILITY
10	0.048
20	0.114
30	0.200
40	0.287
50	0.384
60	0.499
70	0.658
80	0.812
90	0.950
100	0.993